2011

Application of Stochastic Diffusion for Hiding High Fidelity Encrypted Images

Jonathan Blackledge
Technological University Dublin, jonathan.blackledge@tudublin.ie

AbdulRahman Al-Rawi
Technological University Dublin, abdulrahman.alrawi@gmail.com

Recommended Citation

Blackledge, J. M., Al-Rawi, A I. (2011) Application of Stochastic Diffusion for Hiding High Fidelity Encrypted Images. *ISAST Trans on Computing and Intelligent Systems*,2011, vol. 3, issue 1, pages: 46 - 64. doi:10.21427/D7Q916

Application of Stochastic Diffusion for Hiding High Fidelity Encrypted Images

Jonathan M Blackledge and AbdulRahman Isam Al-Rawi

Abstract— **Cryptography coupled with information hiding has received increased attention in recent years and has become a major research theme because of the importance of protecting encrypted information in any Electronic Data Interchange system in a way that is both discrete and covert. One of the essential limitations in any cryptography system is that the encrypted data provides an indication on its importance which arouses suspicion and makes it vulnerable to attack. Information hiding of *Steganography* provides a potential solution to this issue by making the data imperceptible, the security of the hidden information being a threat only if its existence is detected through *Steganalysis*. This paper focuses on a study methods for hiding encrypted information, specifically, methods that encrypt data before embedding in host data where the 'data' is in the form of a full colour digital image. Such methods provide a greater level of data security especially when the information is to be submitted over the Internet, for example, since a potential attacker needs to first detect, then extract and then decrypt the embedded data in order to recover the original information.**

After providing an extensive survey of the current methods available, we present a new method of encrypting and then hiding full colour images in three full colour host images with out loss of fidelity following data extraction and decryption. The application of this technique, which is based on a technique called 'Stochastic Diffusion' are wide ranging and include covert image information interchange, digital image authentication, video authentication, copyright protection and digital rights management of image data in general.

Index Terms— **Image data encryption, information hiding, Steganography, Stochastic Diffusion, encrypted digital watermarks**

I. Introduction

WITH rapid improvements in computer networks and their use for transmitting digital media (such as digital image, audio and video data, for example) coupled with the rapid growth of Internet connectivity, the demand for securing data exchange over the Internet has become increasingly important. The transmission of data over a variety of networks and the internet-based dissemination of digital information has brought about several security issues. Securing data on the Cloud, for example, has become a principal theme for Cloud computing in general. Copyright protection of digital data, copying, editing and the illegal distribution of digital media (such as audio and image data) and the interception of transmitted information by unauthorized parties are common problems requiring innovative and novel solutions. A common solution to these security issues is the use of Cryptography. There are numerous cryptosystems currently available [1] that are commercially available for encrypting data before transmitting it over a network or via the Internet. These systems have been considered computationally secure and are relatively difficult to break. However, although there are many encryption algorithms and systems available [2]-[12], using cryptographic methods alone may not assure the security of a transmission. This is due to a number of reasons. First, the meaningless and randomized form of encrypted data makes interceptors suspect the importance of the information that is being conveyed thereby leading to a potential attack. In addition, rapid increases in the computational performance and sophistication of attack strategies can threaten the security of encrypted data communication systems. Finally, encrypted data may be incriminating in countries when encryption is illegal.

In this paper, an information hiding concept is proposed to reduce the risk of using cryptography only. Data hiding techniques embed information into another medium making it imperceptible to others, except for those who meant to receive the hidden information. Two types of information hiding are available: (i) *Steganography* where the existence of the embedded information is unknown so that interceptors or other unauthorized recipients of the data will not suspect the existence of hidden data; (ii) *Watermarking* where the information is embedded into a cover medium to protect the copyright and/or for the purpose of authentication or to track and trace the original source of information. This paper focuses on studying methods of encrypting hidden information in which cryptographic algorithms are combined with the information hiding methods to increase the security of the transmitted data. In such schemes, the secret data is first encrypted and then embedded into cover data to generate stego-data. The stego-data is then sent through a network or stored online. The unauthorized recovery of hidden encrypted data is very difficult because it needs the attacker or any unauthorized user to detect the existence of the hidden data, extract it from the host data and then decrypt it to recover the original information.

II. Cryptography and Information Hiding

We provide a brief overview of the principles associated with data encryption and information hiding.

Manuscript completed in February, 2011. This work is supported by the Science Foundation Ireland

Jonathan Blackledge - http://eleceng.dit.ie/blackledge - is the Science Foundation Ireland Stokes Professor of Information and Communications Technology and Director of the Information and Communications Security Research Group (ICSRG) at Dublin Institute of Technology - http://eleceng.dit.ie/icsrg. AbdulRahman Isam Al-Rawi - abdulrahman.alrawi@gmail.com - is a PhD research student studying in the ICSRG under the supervision of Professor Blackledge and a Lecturer in the College of Applied Studies, University of Bahrain.

A. Cryptography

Cryptography is the art and science of protecting information (plaintext) by transforming (encrypting) it from a readable state into an unreadable form, (ciphertext) using a secret code or 'key'. Only those who possess the key can decipher (decrypt) the message into the original plaintext form. A cryptographic algorithm is the mathematical function used for encrypting and decrypting the secret data, a one-way function that is not invertible. In general, encryption systems operate on the basis of using two functions. For encryption, we may consider the generic equation

$$E_K(P) = C$$

where E denoted the encryption function, K denotes the encryption key upon which the encryption function and its output critically depends, P denotes the plaintext and C represents the ciphertext. The second (decryption) function D is used for decryption, i.e.

$$D_K(C) = P$$

The encryption/decryption functions along with the plaintext, ciphertext and key(s) form the basis of all cryptosystem. Modern cryptosystem algorithms can be categorized according to the key used for encryption/decryption into private (symmetric) key cryptography and public (asymmetric) key cryptography. Private-key cryptosystems use the same key for encryption and decryption in which the decryption key is the same or can at least be computed from the encryption key, requiring that the sender and receiver agree on one key before communicating with each other. Given that the algorithm is cryptographically secure, the security of such systems relies on the key. The exposure of the key makes it possible for anyone to encrypt and decrypt messages. There are two types of symmetric key algorithms: stream algorithms (stream ciphers) which encrypt only one bit of the plaintext at a time and block encryption algorithms (block ciphers) which encrypts a group of bits on block at a time [13].

Public-key cryptosystems use two different keys for encryption and decryption in which the decryption key cannot be computed from the encryption key. In such systems, the encryption key (public key) can be made public which enables anyone (including an intruder) to encrypt the message, but only a specific recipient with the corresponding decryption key (private key) can decrypt the message. The generic equations for this encryption protocol are as follows:

$$E_{K_{public}}(P) = C$$

$$D_{K_{private}}(C) = P$$

In some cases the sender can encrypt the message using the private key and the receiver decrypts it using the public key Cryptographic methods and their application for securing various types of information are extensively studied in the literature. Such methods include standard algorithms such as DES (Digital Encryption Standards) and DES3, the Advanced Encryption Standard (AES) [13], [14], scrambling methods [15]-[19], chaotic mappings [20]-[25], pseudorandom number generators and stream ciphers [13],[26] and [27].

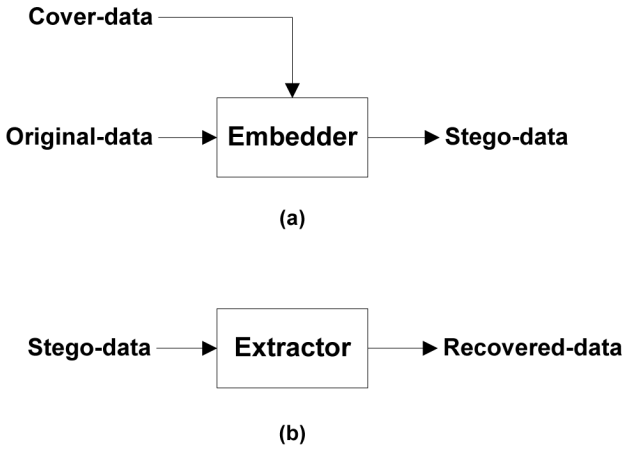

Fig. 1. Information hiding.(a) The process of embedding data into cover data of covertext to generate stego-data or stegotext. (b) The process of extracting the original data from the stego-data.

B. Information Hiding

Information hiding (or 'data hiding') is the process of embedding data into a similar or different form of data so that the hidden information is protected from unauthorized access. The term 'hiding' used here can refer to either making the information imperceptible (as in watermarking) or keeping the existence of the information secret (as in Steganography) [28]. Information hiding techniques can be a good substitute to cryptosystems in situations when the use of data encryption is not feasible or when encryption cannot assure data security due to the encrypted information arousing suspicion or be incriminating in countries where encryption is illegal. Information hiding techniques embed data (secret message) that one wishes to send secretly via an innocuous message (cover/host message) generating the stego-message. The stego-message should be in a form that restricts detection or recovery of the hidden data [29]. The There are two principal data hiding categories, namely Steganography and Watermarking. General information hiding models are described in [29] and Figure 1 illustrates the overall concept.

Steganography is the 'art' of embedding secret messages into other messages in such a way that no one, apart from the sender and intended recipient, suspects the existence of the secret messages [30]. The more recent use of this concept has emerged with the rapid development and communication of digital images, videos, and audio files which can be used as a medium for embedding important data. In other words, we currently live in a covertext rich environment. The main advantage of steganography is that messages do not attract attention to themselves and an examination of the data does not immediately reveal the existence of hidden information. Thus, with regards to a single communication protocol, the user sending the hidden data and the recipient of the data are the only ones who know about the existence of the hidden data [31]. Many steganographic techniques have been proposed in the literature [29] but steganography can be categorised into three basic types: (i) pure steganography in which the sender embeds the secret data directly and the receiver extracts it

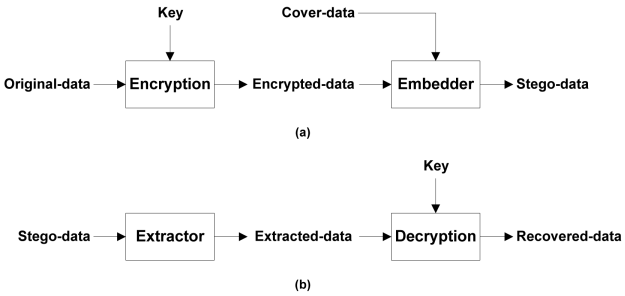

Fig. 2. Encrypted Information hiding. (a) The process of encrypting data and then embedding it into a cover data to generate the stego-data. (b) The process of extracting the secret data from stego-data and decrypting it to recover the original data.

likewise; (ii) the private-key steganography where the sender uses a private key to embed the secret information in a way that is similar to private-key encryption; (iii) public-key steganography, in which the sender embeds the secret data using a private-key and the receiver extracts it using a public-key, this process is similar to the public-key cryptosystems [32].

Watermarking is the process of embedding information into another medium in a way that is difficult to remove which is usefull to protect the source of the information and avoid copyright violation, for example. An example of this concept is visible watermarking where the watermark is visible in the media used such as a text or logo which identifies the owner of the media. Other classes of watermarking include invisible watermarking, in which the information is added to the media in such a way that it cannot be recognized. Digital watermrking systems can be further categorized according to the robustness of an attack into fragile, semi-fragile and robust. Watermarking can be used for copyright protection, source tracking or covert communications.

C. Encrypted Information Hiding

Compared with information hiding in general, there are relatively few publications that have addressed the issue of hiding encrypted information. As in conventional information hiding methods, Encrypted Information Hiding (EIH) systems can be categorized into two types: spatial domain systems and frequency domain systems. Figure 2 illustrates the principle associated with EIH. Spatial domain EIH is applied in the spatial domain, in which the original image pixels (in the case of a digital image covertext) undergo direct modification. Those systems are generally considered simple and often require less computational cost, although they can be less robust against attacks such as filtering and cropping [33]. Frequency domain EIH is performed by transforming the covertext image into the transform domain and changing its coefficients to embed information. These systems require more computational cost and are generally considered more complex to implement, although they are also considered to be more robust.

III. Encrypted Information Hiding in the Spatial Domain

We now provide an overview of the published literature available that is based on EIH in the spatial domain. In [34], [35] the authors present a new approach for hiding encrypted information into a digital image. Their approach is based on the concept of stochastic confusion and diffusion. The proposed system generates a maximum entropy cipher using a pseudorandom number generator (PRNG) or an appropriate encryption method. The cipher is convolved with a secret image using stochastic diffusion and the result is quantized to a 1-bit array and embedded into a host image. The encrypted data can be embedded into the lowest 1-bit layer or into multiple 1-bit layers of the host image. To recover the encrypted information, the hidden data is extracted from the 1-bit layer(s) of the host image and the result decrypted by correlating it with the original cipher. The convolution/correlation processes are undertaken in the Fourier domain. The proposed method can operate on 24-bit color images as well. The original color image is decomposed into the Red, Green and Blue channels. Each channel is encrypted by convolving it with a cipher (the same or a different cipher can be used for encryption) to generate three encrypted channels. These channels are then embedded into the three-channels of a 24-bit color cover image. Data recovery is carried out by extracting and decrypting the data from each channel separately and then combining the three channels to form the recovered image.

X Li et al. [36] propose a novel method for the hidden transmission of biometric images based on chaos and image content. To increase the security of the watermark, it is first encrypted using the chaotic map where the key(s) used for encryption are derived from a palm print image based on the pixel value distribution, illumination associated with various image distortions which are different for each palm print image (even if they are for the same person). The authors used the normalized mean value of three randomly selected pixels from the palm print image as an initial condition for the chaotic map. The logistic map is used to generate a one-dimensional sequence of real numbers which are mapped into a binary stream to encrypt the watermark using an XOR operation. The encrypted watermark is embedded into the same palm print image used to derive the secret key(s). The stego-palm-print image is hidden into the cover image using a novel content-based hidden transmission scheme proposed by the authors. First the cover image is segmented into different regions using a classical watershed algorithm. Due to the over-segmentation result of this algorithm, a RFCM (Region-based Fuzzy c-means Clustering) algorithm is used to merge similar regions. The entropy of each region is calculated and the stego-palm-print image is embedded into the cover image according to the entropy value where more information is embedded in highly textured regions than uniform regions. A threshold value T is used to partition the two regions. If the entropy is greater than T, the binary streams of the secret data are inserted into the 4 least significant bits of the region and if the entropy is smaller than T, the binary streams of the secret data are inserted into the 2 least significant bits of the region. Color cover

images are decomposed into RGB channels before embedding. J Kong et al. [37] present a novel method of transmitting hidden biometric images based on chaos and image content that is similar to [36]. A secret grayscale image of size 128×128 is converted into a binary stream and then encrypted using a logistic mapping, the encryption keys being generated randomly using any pseudorandom number generator. A color cover image of size 256×256 is converted to a grayscale image, segmented into different regions using a conventional watershed algorithm. A fuzzy c-means clustering algorithm is then used to merge similar regions, each region being classified into certain clusters except for those regions of the watershed lines. The k-Nearest Neighbour (KNN) algorithm is used to partition regions needing re-segmentation. For the resultant image without watershed lines, the entropy is calculated and the secret image is embedded according to the entropy values. The color cover image is decomposed into RGB channels and highly textured regions are used to embed more information. A threshold value T is used to separate the two regions. If the entropy is smaller than T the binary streams of the secret data are inserted into the 2 Least Significant Bits (LSB) of the three channels of the region. If the entropy is greater than T, the binary streams of the secret data are inserted into the 4 LSB of the three channels of the region.

C W Lee and W H Tsai [38] proposed a Steganographic method using PNG formatted images based on an information sharing technique. The secret image M is divided into shares using a (k, n)-threshold secret sharing algorithm where secret shares are embedded into the alpha-channel of the PNG cover image. The hiding procedure can be summarized as follows: Divide *M* into *t*-bit segments with *t*=3 and transform each segment into a decimal number resulting in a decimal number sequence where a *(4,4)*-threshold secret sharing algorithm is used to generate the partial shares *F*; embed *F* into the cover image *I* by replacing its alpha-channel values with the values of *F*. The process is repeated for all decimal values of the secret data resulting in a stego-image *I'*. In general, if every four *t*-bit segments are transformed and embedded similarly, then $R = (4 \times t) \times (S/4) = tS$ bits where S is the size of the image. This means that the data hiding capacity is proportional to the chosen value of *t*. However, the larger the value of *t* the lower the visual quality of the stego-image since this causes a wider range of the alpha-channel values to be altered, leading to a more obvious non-uniform transparency effect appearing on the stego-image. The value of *t* is selected to ensure the uniform distribution of the stego-image's alpha-channel.

C Uuefen et al. [39] introduce the principle of image scrambling and information hiding and proposed a double random scrambling procedure based on image blocks. A secret image of size $M \times N$ is divided into small sub-blocks (e.g. 4×4 or 8×8 blocks) and a scrambling algorithm using *key1* carried out to randomize the sub-blocks. However, the information in each inner sub-block remains the same as the original version and so another scrambling algorithm with *key2* is utilized to destroy the autocorrelation in each inner sub-block and to increase the difficulty in decoding the secret image. To make the hidden secret image more invisible, its histogram is compressed into a small range. Hiding is then performed by simply adding the secret image to the cover image. The extraction of the original image can be summarized as follows: extract the embedded information from the host image, expand the histogram of the extracted image and then apply a descrambling process among the sub-blocks and inner sub-blocks, the original secret image thus being reconstructed.

M K Kundu and S Das [40] present a watermarking scheme that combines lossless compression and encryption for applications to medical images. They proposed that a doctor/radiologist is interactively provided with a defined polygonal ROI. This is characterized by the number of vertices in the polygon n_v and the vertex coordinate *v(x,y)*. A SHA-256 hashing algorithm is then used to calculate a one-way hashing value for the ROI denoted by $HASH_{ROI}$. The data is encrypted using the Advanced Encryption Standard (AES) algorithm denoted by PI_{ENCRY}. The user then specifies the bit-plane of the ROI bits represents by BIN_{BP}. A watermark WM_{CONCAT} is generated by concatenating n_v, the vertex co-ordinate *v(x,y)*, $HASH_{ROI}$, PI_{ENCRY} and BIN_{BP} which is then compressed from WM_{CONCAT} to WM_{COMP} using an arithmetic integer compression method. The compressed data is then converted to a binary string WM_{EMB} which in turn is embedded in the image using the bit-plane specified by the user. In order to increase the security, WM_{EMB} is embedded randomly using any pseudorandom number generator algorithm. Watermark extraction and verification is accomplished by first extracting WM_{EMB}, decompress it, separate the concatenated values, calculate a hash value $HASH_{ROI-EXT}$ using the same hashing algorithm and comparing the result with $HASH_{ROI}$. If both values are equal then the image is authenticated and PI_{ENCRY} is decrypted using the same key used for encryption. Otherwise the watermarked image is not authenticated and rejected.

Y. Lu, et al. [41] present a novel lossless and content-based hidden transmission method for biometric images. Firstly, the watermark is encrypted using a chaotic mapping, the secret keys being generated from a biometric (*palmprint*) image and used as a parameter value and initial condition for the chaotic map. The encrypted watermark is embedded into the palm-print image using a lossless data hiding scheme. Finally, the stego-palm-print image is embedded into the cover image using a content-based steganographic scheme as described in [36]. The extraction is performed by reversing the embedding process, the extracted watermark will being used to authenticate the image content. If the extracted and the original watermark similarity satisfies a pre-defined threshold, then the palm-print image is accepted, otherwise it is invalidated.

M. Sabery and M. Yaghoobi [42] introduce an image hiding method to hide a grayscale image into a color image using chaotic return maps. The user initially inserts a key which is converted into binary form to be utilized as an initial condition for the logistic map. The chaotic mapping is used to identify the image pixels location (*x,y*) to hide the secret image. The last three bits of the Red and Green channels, and the last two bits of the Blue channel are set to zero and grey level values of the secret image represented in terms of a one-dimensional matrix $V = \{v1, v2, ..., v_{MM}\}$ row by row. Finally, the bits v_{i1}, v_{i2}, v_{i3} of the matrix V are added to the last three bits

of the Red channel, the bits v_{i4}, v_{i5}, v_{i6} are added to the last three bits of Green channel and the bits v_{i7}, v_{i8} are added to the last two bits of Blue channel at the position (*X,Y*) of the cover image.

J. M. Blackledge [43] presents a novel approach for hiding encrypted data in a digital image. The secret image is encrypted with a key using the process of stochastic diffusion. A random number generator is used to generate a noise field which is then convolved with the secret image. The encrypted output is quantized into a 1-bit array to generate a binary cipher, which is embedded into the lowest 1-bit layer of a host image. The extraction process is performed by extracting the binary cipher from the host image and correlating it with the original noise image.

D. C. Lou and J. L. Liu [44] propose a steganographic method for secure communications by using the concepts of cryptography and information hiding. The idea behind the proposed method is to use the variant-size LSBs and the insertion of redundant Gaussian noise into the cover image to survive a cover-carrier attack. The cover image used is a grayscale image. First, the secret image is compressed and encrypted using a session key. The session key is encrypted with the receiver's public key and the result used to seed the generation of random variables with a bounded normal distribution. The variables generated are of size $M \times N - l$ (where M and N denote the row and column size of the cover image, respectively). The random numbers conform to a redundant Gaussian noise field and are used to determine the number of pixels' LSBs for embedding the secret data. The seed and the size of the secret data are embedded into the last *l* pixels of the cover image and the random numbers are divided into two parts according to a secret mixing rate; the first part is used as a travelling order for embedding the seed and the size of the secret data; the second part is used as a traveling order for embedding the Gaussian noise. The encrypted data is embedded into the remaining pixels of the cover image and the random numbers are used to embed the secret data and to determine the number of pixels' LSBs used for embedding.

Z. Liu, M. A. Ahmad and S. Liu [45] present an image sharing scheme based on a combination theory. The image is first encrypted by multiplying it by random matrices and them, based on a (*t,n*) threshold secret sharing scheme, the encrypted image is shared into *n* shadow images which are of the same size as the original secret image. The shares can be hidden using any information hiding method. To reconstruct the secret image, at least *t* shares should be combined to retrieve the original data. If the secret image is a color image, it is decomposed into RGB channels and each channel is treated as a grayscale image to which the same method is applied. In general, any encryption and sharing algorithm can be utilized to encrypted and generate the shadow images, respectively.

In [10] the authors propose a virtually imperceptible image hiding scheme based on vector quantization (VQ). Their goal is to design a high quality and a high capacity image hiding scheme which is based on the VQ compression and a Digital Encryption Standard (DES) based cryptosystem. In the proposed scheme, a *t* 8-bit gray level secret image of size $w \times h$ is embedded into a cover image to produce the stego-image as follows: Firstly, the Linde-Buzo-Gray (LBG) algorithm [47] is used to generate a VQ codebook *C* of size N_c with codewords of size $m \times n$. Each secret images is then partitioned into blocks of $m \times n$ which are then encoded into a binary indexed codeword. The indices are merged to obtain a compressed message for all secret images, the compressed information being embedded into *C* by LSBs substitution to generate *C*. Finally, *C* is encrypted using the DES algorithm and embedded into the cover image to generate the stego-image. The extraction procedure is performed by decrypting the modified codebook *C* and the compressed information extracted using parameters that include the number of secret images *t*, the size of the secret image $w \times h$, the codebook size N_c and the codeword size $m \times n$.

In [48] a new steganographic method for hiding grayscale or color secret images in a true color image is proposed. The approach makes use of the color quantization technique to make the hiding procedure independent of the image type, and utilizes the DES encryption algorithm to increase the security of the secret information. Color, palette-based 256-color, or grayscale images can be embedded in the host image and annotation data can be embedded if necessary. In the case of color images, quantization of a 256-color palette is applied using a LBG clustering algorithm [47]. After obtaining the color table, each pixel's color of the secret image is replaced by the nearest color from the color table using a Euclidean distance metric to obtain the quantized 256-color secret image. Color palette, annotation data and indices of the secret images pixels are represented in binary form and encrypted by using the DES algorithm. The host image boundaries are selected to hide the color table and the annotation data as follows: The successive 512 color pixels located in the *third* row from the top boundary and the first successive 256 color pixels located in the *third* row from the lower boundary are selected to hide the color palette table. The remaining 256 color pixels located in the *third* row and the *fifth* row from the lower boundary and the *fifth* and *seventh* rows from the top boundary are selected to embed the annotation data. The rest of the color pixels for the host image are used to hide all of the 8-bit indices of pixel data from the secret image. If the secret image is a 256-color image, the color quantization step is skipped, and if it is a grayscale image, then each grey level value is directly embedded into the corresponding value of the true color cove image. The inverse procedure is performed by extracting the encrypted palette data, image data and the annotation data, decrypting them and finally constructing the quantized secret image.

In [49] a prediction-based image hiding scheme is presented, which utilizes the concept of data compression and encryption for information hiding. The proposed method embeds secret data into the compression code during image compression. The secret image is encrypted using the DES algorithm. The host image pixels are scanned from top to bottom and from left to right, then a modified Median Edge Detector (MED) predictor is used to estimate the Predictive Pixel Values (PPVs) for the host image pixels (original MED prediction-based image coding is explained in [50]). The prediction Error Value

(EV) is calculated by computing the difference between the Original Pixel Values (OPVs) and the PPVs. The encrypted secret image is sequentially embedded into each prediction EV based on entropy coding to produce the secret compression codes, which form the hiding result. To extract the hidden information, the prediction pixel values are extracted from the received compression codes by decoding them based on the same entropy codec, values which are then used to extract the encrypted secret image which is then decrypted using the DES algorithm to obtain the original secret image [49].

In [51] an image hiding method scheme with modulus function and dynamic programming on partitioned pixels is proposed. The secret image pixel values are transformed into other values based on substitution tables following a dynamic programming strategy. The function of the Substitution Table (ST) is to determine to which *k*-bits values each secret *k*-bits unit should be transformed. Since different substitution tables can be produced for the host image, an optimal table is selected to minimize the degradation of the stego-image. The method given in [52] is used to find the optimal substitution table. The host image pixel values are partitioned into two groups, G_L and G_U, using a threshold value *T*. The number of secret image bits that can be embedded into the host image is 3 for host image pixels belonging to G_U and 2 for host image pixels belong to G_L. Before embedding, the secret image pixel values are transformed using optimal STs. The proposed scheme uses two optimal STs, one for the host image pixels above *T* denoted by ST_U and the other for the host image pixels below *T* denoted by ST_L. A dynamic programming strategy is employed to transform the corresponding secret image pixels, which are then embedded into the host image based on modulus functions (two modulus functions M_U and M_L are used according to the pixels group) to obtain the stego-image. The extraction procedure is employed by checking the stego-image pixel value, if it is larger than the threshold value *T*, M_U is used to extract the hidden data, otherwise M_L is used. The original secret image is recovered by transforming the extracted pixel values according to the optimal substitution tables ST_L and ST_U.

In [53] a method for sharing and hiding secret images is presented. The proposed method is divided into three stages. The first stage involves image quantization, in which the secret image is quantized using a Variable Size Quantization (VSQ) sub-procedure; the secret image is divided into blocks of size 1×2 which are then examined to determine smooth and edge blocks. The edge blocks are quantized using an edge quantization method while smooth blocks are merged to form blocks of size 1×4 which are then quantized using a smooth quantization method. A table is generated to keep track of edge and smooth blocks by recoding 1 for edge blocks and 0 for smooth block. This table is embedded into the quantized image rather than attaching it to the quantized image to avoid increasing its size; the output at this stage is the quantized-embedded image. The second stage is the image sharing stage; the quantized-embedded image is shared by applying a (*t,n*) threshold scheme which generates *n* shadow images which appear to be random noise when only *t* ($t \leq n$) of them are required to recover the original image. The authors slightly modified the method in [54] by changing the range of each shadow image pixel value from 0-255 to 0-16 which is the same range of the quantized-embedded image. This modification makes it possible to embed the shadow image into an ordinary host image of the same size. In the third and final stage, the *n* shadow images are embedded into *n* host images using the hiding method described in [53] to generate *n* stego-images, although, in general, any hiding method can be applied. The recovery of the original secret image is performed by retrieving any *t* or more of the stego-images, extract the hiding shadow images and combine them using the same method proposed in [54].

In [55] a new multiple image encryption and watermarking method is proposed. The color host image is first decomposed into three RGB channels and each channel is enlarged to four times its original size based on [56]. The three grayscale images obtained from the RGB channels are encrypted using a fractional Fourier transform (FRFT) and a region shift encoding method. The three encrypted watermarks are embedded into the three RGB channels of the host image using the method given in [55] which is based on the information hiding and extraction method proposed in [57] and [58]. The decryption process is based on an inversion scheme where the original watermark images cannot be recovered without revealing the FRFTs two fractional orders used to encrypt each watermark in addition to the random matrix used in the region shift encoding method.

In [59] a new type of encoding methods to hide information in a host image is proposed. The types of the covert data proposed are *plot*, *fax*, *word*, and *network data*. The method can be summarized as follows: (i) Let *H* of size $M \times M$ be the host image that is to be modulated to generate the overt image *H**; (ii) Create a new array *R* with *N* elements where *R* elements consist of the identification codes, types code and the rest are either 0 or 1 and encode the covert information as a binary data string (codes) as described in [59] (Section 2.1.3) according to covert information of type (*plot, fax, word, or network data*); (iii) Copy the elements of the binary data string into a new matrix *S* of size $(M-1) \;\; N$ row-by-row and from left-to-right and then copy the arrays *R* and *S* into a new a matrix *T* of size $M \;\; N$ where the first row of *T* is copied from *R* and the rest are copied from *S*. The host image *H* is modulated to the overt image *H** by subtracting *T* from *H* (i.e. $H* = H - T$). The overt images *H** pixels are classified into three groups. The first group contains the identification codes which are used to determine if *H** is produced. Those codes that contain random binary bits are recognized only by the recipient of the data. The second group contains type codes to determine the embedded information and are given by 00, 01, 10, and 11 for *plot, fax, word, and network data*, respectively. The last group contains the encoded information which is used to decode the encoded data.

The authors in [60] and [61] present an approach to digital watermarking based on the use of cryptography in conjunction with watermarking. The watermark in encrypted by diffusing it with a cipher to produce a scrambled image. The cipher is a noise field that can be generated using any conventional random number generator or chaotic mapping. The scrambled

image is then 'confused (an additive process) with a host image to hide the encrypted image and produce to the stego-image. This method can be applied to any e-to-e digital image transmission because it is relatively insensitive to lossy compression. The authors proposed the use of 'diffusion only watermarking for printed document authentication. The reason for this is that the 'print/scan cycle generates an array of pixels that are different to the original image (even if they are both look similar) including the image size, its orientation and pixel intensity. This is because the pixels positions associated with the cover image cannot be assured when it is printed and scanned (i.e. the print/scan cycle leads to pixel de-registration). The watermark image is diffused with the Point Spread Function (PSF) that is characteristic of the print/scan cycle which enables the receiver to recover the watermark data subject to a degradation characterized by the PSF. This method can be carried out in color prints by applying exactly the same technique to the RGB channels.

IV. Encrypted Information Hiding in the Transform Domain

D.W. Kim et al. [62] presente an approach for hiding a digital hologram (DH) using a conventional encryption algorithm (AES-128). This work is devoted mainly to the experimental effects of encryption on information hiding for digital holograms in three data domains: the hologram domain, the DCT (Discrete Cosine Transform) domain and the DWT (Discrete Wavelet Transform) domain. For the hologram domain, and due to energy calculation experiments, only encryption of the three most significant bit plane could hide the information even if the remaining bits are exposed. This is because 90% of the total energy represents the three bits and 25% of the data needs to be encrypted to effectively hide the information in the hologram domain. Transforming the hologram image into DCT blocks of different sizes and calculating the energy for all DC coefficients shows that scrambling only the DC coefficients is sufficient to hide the information of the content [62] and that the energy ratios of DC coefficients in all segmentation cases are more than 95% of the total energy (in segmentation sizes of 128×128). Only 0.0061% of the data is encrypted which is more efficient than in hologram domain case. In the DWT domain, the experiments show that encrypting part of a particular sub-band is not useful and that a whole sub-band needs to be encrypted. According to the results given in [62] (Table 3), at least the first sub-band should be encrypted to hide the content unrecognizably (i.e. 50% of the data needs to be encrypted). The results show that the DCT domain provides the best domain because the encryption ratio about 0.00001.

In [63] a steganographic-based approach is used to protect the iris code data by hiding it in a digital image for personal data identification purposes. The logistic chaotic map with the first key is applied to encrypt the iris code (extracted iris features) into a chaotic sequence of length *N*. The host image is divided into 8×8 blocks and the DCT coefficients for the k^{th}, denoted by B_k, is then transformed into a one-dimensional sequence D_k by the means of 'zig-zag scanning. A sequence of real numbers $X_i, \quad i = 1, 2, ...N$ is generated using the logistic chaotic map with the second key. By locate the embedding location of each block based on the previously generated sequence, the positions in the mid-to-high frequency band in each block is selected as possible embedding points. The actual embedding position sequence $P_i, \quad i = 1, 2, ...N$ is obtained by transforming the sequence X_i using the mapping $P_i = X_i \times l + s$, where l is the length of the selected interval and s is the selected start position. The encrypted iris code is then hidden using the method given in [40], each block is data inverted using the zig-zag order D_k back into two-dimensional blocks and the inverse DCT applied. The new blocks generate are an approximation to the host image including the secret information (the stego-image). The extraction process is performed by transforming the stego-image into the DCT domain, acquiring the one-dimensional sequence of DCT blocks using the zig-zag order to obtain the embedding positions using the same chaotic sequence used for hiding and selecting the correct coefficients to extract the secret information. The encrypted one-dimensional sequence is the decrypted to obtain the original data the iris code.

In [64] a steganographic method based on JPEG compression is proposed. The scheme uses the concept of cryptography and data compression. This article is similar to the work presented in [65] with improved hiding capacity and stego-image quality. The proposed method consists of five stages. In the first stage, the secret data *M* (text, image, or video) is encrypted using any conventional cryptosystem to obtain the vector S, where $S = s_1, s_2, ..., s_m$ and s_i is a secret bit stream (with values 0 or 1) and m is the length of S. In the second stage, the cover image I is partitioned into 8×8 pixel blocks and each block transformed into DCT coefficients by applying the DCT transform. The DCT coefficients are quantized using a modified JPEG quantization table. In the third stage, the encrypted data S is embedded into the mid-frequency component of the quantized DCT coefficients for each block; each DCT coefficient is embed using two bits of the secret data in the least two significant bits. In the fourth stage, the modified DCT coefficients are compressed using JPEG entropy coding and the JPEG stego-image E generated. Data recovery is based on applying JPEG entropy decoding to E, extracting the secret data bits from the quantized DCT coefficients and decrypting the data.

In [66] a new hybrid scheme based on a Singular Value Decomposition (SVD), norm quantization and a Modulo-2 approach is considered. Two separate procedures are used for embedding two binary watermark images into a grayscale image of size 256×256. For the first level, the first binary watermark image of size 32×32 is encrypted using a chaotic scrambling algorithm and a SVD matrix norm quantization method applied to the host image. The encrypted watermark is then embedded into the host image decomposition. For the second level, the low frequency sub-band from the first level is decomposed using the IWT-Modulu-2 method resulting in four sub-bands of size 64×64. The second binary watermark is then embedded into the second level decomposition.

In [67] an adaptive digital watermarking algorithm based on chaos and image fusion is presented. The watermark image

is first encrypted using the logistic chaos map. The adaptive watermark embedding algorithm is then applied based on NVF (Noise Visibility Function) and image fusion, which is used in the wavelet domain. When the host image and the watermark are fused, the boundary values are selected to be small or large depending on the image sensitivity to the noise. The NVF value (the proposed algorithm chooses $1 - NVF$) are therefore utilized as a boundary value. Fusing the original image and the watermark using fixed fusion (multiplication) factors can lead to two problems: (i) the fused image is easily attacked; (ii) the human visual perception model is not fully considered. The proposed algorithm chooses $(1 - NVF) \div 2p$ as a multiplication factor for the watermark image and $1 - (1 - NVF) \div 2p$ as a multiplication factor for the host image where p is used to adjust the 'invisibility permission. To insure watermark recovery, the watermark image is multiplied by a constant α which magnifies the wavelet permission for image watermarking based on cryptic fusion method and the NVF.

In [68] an approach to encrypting and hiding a gray-scale image using a wavelet packet transform and bit plane decomposition is proposed. First, the secret image is encrypted by applying a chaotic mapping, then the encrypted image is decomposed into eight bit planes, accomplished separately, taking the highest bit of every pixel from the secret gray-scale image (consisting of 8 bits) to compose a new binary image B7, the other bit plans (B6-B0) being similarly composed. The cover image is decomposed using a wavelet packet transform into sixteen sub-band images and eight high frequency sub-images selected from them. The entropy value of the high frequency sub-images are calculated and sorted in descending order (E7-E0). The bit planes (B7-B0) are hidden separately in the appropriate sub-images (E7-E0) and the wavelet packet applied to reconstruct the cover image with the secret information embedded into it. This hiding method does not change the information sequence of the sub-images which makes it robust to low-pass filtering and image compression.

In [69] an information hiding algorithm based on double uni-dimensional chaos sequences and a non-uniform B-spline wavelet is proposed. The secret image is encrypted by applying a double logistic chaos mapping. The first mapping is applied to encrypt the image gray values (pixel values) and the second one is applied to encrypt the image pixel positions. The cover image *C* is transformed to a non-uniform B-spline wavelet domain, the transformed image being recorded as coefficient matrix *E*. The secret image is inserted in the most important parts of the image (i.e. the low frequency areas) in order to increase invisibility and image robustness. Appropriate coefficients required to hide the secret data is undertaken by constituting the coefficient vector $e(i)$, embedding the secret information by adjusting $e(i)$ to $e(i)$, and then inserting $e(i)$ back to the matrix *E* to reconstruct the embedded image *C*.

In [9], a new watermarking method for covert communication and copyright protection is presented. The method is based on a visual model to calculate the block based Just Noticeable Difference (JND), the block discrete wavelet transform (DWT), redundant encoding and chaotic scrambling. An image block B_{uv} is selected each time from the cover image *I* and transformed using the DWT. Three sub-band images are used for embedding. The JND value of the DWT sub-bands for each block is calculated to ensure transparency and robustness of the watermark. The watermark (text or image) is converted into a binary sequence W. In order to ensure that the length of the original watermark is equal to $1 \div 4$ of the total pixels of the cover image [9], zeros padding is applied. The watermark is extended W_{ex} to ensure its redundancy which is important to improve its robustness. Chaotic scrambling is applied to W_{ex} to increase security which is then embedded into the cover image. Finally an inverse DWT is applied to all the watermark embedded blocks in order to construct the stego-image *I*. Reconstruction is performed by extracting each block B_{uv} watermark to obtain W_{ex}. Decryption is applied (as necessary) and the watermark W obtained from W_{ex}.

In [71] a chaotic secure content-based hidden transmission of biometric templates is proposed. This method is based on the concept of encryption and data hiding. The input image (iris biometric data of the user) is captured and the important iris features extracted based on an image processing method described in [72]. The extracted feature vector is taken to be the secret data to be transmitted. The iris feature is encrypted using double chaotic maps, the parameter value and the initial conditions (the bio-keys) being required to generate the chaotic map which are derived from the original iris template. The first chaotic map - a logistic mapping - is used to generate a one-dimensional sequence of real numbers which are used as a sequence key to generate the second chaotic map - the Henon map [73]-[74]. The Henon chaotic map is used to encrypt the iris features. The cover image is transformed into the wavelet domain using a DWT and the encrypted data is embedded into the LH2, HL2 and HH2 sub-bands. The hiding positions are based on a pre-generated pseudo-random number stream. If the size of the iris template is large, then the LH3, HL3 and HH3 sub-bands are also used. Finally the inverse DWT is applied to the host image to reconstruct the stego-image. Template extraction is performed by decomposing the stego-image using DWT, extracting the hidden data using the same random hiding template and decrypting the extracted features which are then compared with the original data for verification.

In [75] a technique for digital image hiding and encryption with a double random phase based on the fractional Fourier transform (FRFT) is proposed. The (color) secret image is decomposed into the RGB channels and each component transformed into the fractional Fourier domain and encrypted using two random functions and four fractional orders of the FRFT. The host image is also decomposed into RGB, each channel transformed into the fractional Fourier domain and the encrypted data embedded into the corresponding host image channel with a weighting parameter being used to control the quality of the stego-image. The extraction procedure proposed in this work uses half-blind watermarking in which the original host image and the keys are required to recover the original secret image.

M. K. Khan, L. Xie and J. Zhung [76] propose a fingerprint template hiding method based on chaos and a Non-uniform Discrete Fourier Transform (NDFT). The captured fingerprint template is preprocessed using a Gabor filter bank-based

technique to extract its important features [77], [78]. The fingerprint template is encrypted using a skew-tent chaotic map [79], [80]. The encrypted template is encoded by BCH coding [81] in order to help correct errors generated by noise during transmission of the hidden information when the template is decoded. The BCH coded fingerprint template is modulated using two chaotic maps. The first map, which is an iterative chaotic map with infinite collapses [82] is used to generate a sequence of real numbers to be utilized as a bifurcating parameter for the second chaotic map the Chebyshev map [83]. This second map is used to modulate the BCH coded template as described in [76]. Finally the encrypted, encoded and modulated fingerprint template is embedded into the selected mid-frequency coefficients (MFCs) of the NDFT domain of a host audio signal. The hiding procedure can be summarized as follows: (i) Segment the cover audio signal into 8 samples per segment; (ii) for each segment, a frequency point in MFCs is chosen based on a secret key generated from the logistic chaotic; (iii) The selected frequency coefficients are quantized to embed the biometric template as given in [76] - equation (11) - to generate to stego-audio signal. For template extraction and matching, the NDFT is taken of the stego-audio to recover the hidden data. The template is then modulated, decoded and decrypted and the output compared with the original stored template using equation (14) in [76].

V. Lossy Encrypted Image Information Hiding using Stochastic Diffusion

In 'image space', we consider the plaintext to be an image $p(x, y)$ of compact support $x \in [-X, X]; y \in [-Y, Y]$. Stochastic diffusion is then based on the following results:

Encryption

$$c(x, y) = m(x, y) \otimes_x \otimes_y p(x, y)$$

where

$$m(x, y) = \mathcal{F}_2^{-1}\left[M(k_x, k_y)\right]$$

and $\forall k_x, k_y$

$$M(k_x, k_y) = \begin{cases} \frac{N^*(k_x,k_y)}{|N(k_x,k_y)|^2}, & \mid N(k_x, k_y) \mid \neq 0; \\ N^*(k_x, k_y), & \mid N(k_x, k_y) \mid = 0. \end{cases}$$

Decryption

$$p(x, y) = n(x, y) \odot_x \odot_y c(x, y)$$

Here, k_x and k_y are the spatial frequencies and $\mathcal{F}_2^{-1}$ denotes the two-dimensional inverse Fourier transform. For digital image watermarking, we consider a discrete array $p_{ij}, i = 1, 2, ..., I; j = 1, 2, ..., J$ of size $I \times J$ and discrete versions of the operators involved, i.e. application of a discrete Fourier transform and discrete convolution and correlation sums.

If we consider a host image denoted by $h(x, y)$, then we consider a watermarking method based on the equation

$$c(x, y) = Rm(x, y) \otimes_x \otimes_y p(x, y) + h(x, y)$$

where

$$\|m(x, y) \otimes_x \otimes_y p(x, y)\|_\infty = 1$$

and

$$\|h(x, y)\|_\infty = 1$$

By normalising the terms in this way, the coefficient $0 \leq R \leq 1$ can be used to adjust the relative magnitudes of the terms such that the diffused image $m(x, y) \otimes_x \otimes_y p(x, y)$ becomes a perturbation of the 'host image' (covertext) $h(x, y)$. This provides us with a way of digital watermarking one image with another, R being referred to as the 'watermarking ratio', a term that is equivalent, in this application, to the standard term 'Signal-to-Noise' or SNR as used in signal and image analysis. For colour images, the method can be applied by decomposing the image into its constituent Red, Green and Blue components. Stochastic diffusion is then applied to each component separately and the result combined to produce an colour composite image.

For applications in image watermarking, stochastic diffusion has two principal advantages:

- a stochastic field provides uniform diffusion;
- stochastic fields can be computed using random number generators that depend on a single initial value or seed (i.e. a private key).

A. Binary Image Watermarking

Watermarking a full grey level or colour image in another grey or colour image, respectively, using stochastic diffusion leads to two problems: (i) it can yield a degradation in the quality of the reconstruction especially when R is set to a low value which is required when the host image has regions that are homogeneous; (ii) the host image can be corrupted by the watermark leading to distortions that are visually apparent. Points (i) and (ii) lead to an optimisation problem with regard to the fidelity of the watermark and host images in respect of the value of the watermark ratio that can be applied which limits the type of host images that can be used and the fidelity of the 'decrypts'. However, if we consider the plaintext image $p(x, y)$ to be of binary form, then the output of stochastic diffusion can be binarized to give a binary ciphertext. The rationale for imposing this condition is based on considering a system in which a user is interested in covertly communicating documents such as confidential letters and certificates, for example.

If we consider a plaintext image $p(x, y)$ which is a binary array, then stochastic diffusion using a pre-conditioned cipher $0 \leq m(x, y) \leq 1$ consisting of an array of floating point numbers will generate a floating point output. The Shannon Information Entropy of of any array $A(x_i, y_i)$ with Probability Mass Function (PMF) $p(z_i)$ is given by

$$I = -\sum_{i=1} p(z_i) \log_2 p(z_i)$$

The information entropy of a binary plaintext image (with PMF consisting of two components whose sum is 1) is therefore significantly less than the information entropy of the ciphertext image. In other words, for a binary plaintext and a non-binary cipher, the ciphertext is data redundant. This provides us with the opportunity of binarizing the ciphertext

by applying a threshold, i.e. if $c_b(x,y)$ is the binary ciphertext, then

$$c_b(x,y) = \begin{cases} 1, & c(x,y) > T \\ 0, & c(x,y) \leq T \end{cases} \quad (2)$$

where $0 \leq c(x,y) \leq 1 \forall x,y$. A digital binary ciphertext image $c_b(x_i, y_j)$ where

$$c_b(x_i,y_i) = \begin{cases} 1, & \text{or} \\ 0, & \text{for any } x_i, y_j \end{cases}$$

can then be used to watermark an 8-bit host image $h(x,y), h \in [0,255]$ by replacing the lowest 1-bit layer with $c_b(x_i,x_j)$. To recover this information, the 1-bit layer is extracted from the image and the result correlated with the digital cipher $n(x_i,y_j)$. Note that the original floating point cipher n is required to recover the plaintext image and that the binary watermark can not therefore be attacked on an exhaustive XOR basis using trial binary ciphers. Thus, binarization of a stochastically diffused data field is entirely irreversible.

B. Statistical Analysis

The expected statistical distribution associated with stochastic diffusion is Gaussian. This can be shown if we consider a binary plaintext image $p_b(x,y)$ to be described by a sum of N delta functions where each delta function describes the location of a non-zero bit at coordinates (x_i,y_j). Thus if

$$p_b(x,y) = \sum_{i=1}^{N}\sum_{j=1}^{N} \delta(x-x_i)\delta(y-y_j)$$

then

$$\begin{aligned} c(x,y) &= m(x,y) \otimes_x \otimes_y p(x,y) \\ &= \sum_{i=1}^{N}\sum_{j=1}^{N} m(x-x_i, y-y_j). \end{aligned}$$

Each function $m(x-x_i, y-y_j)$ is just $m(x,y)$ shifted by x_i, y_j and will thus be identically distributed. Hence, from the Central Limit Theorem

$$\Pr[c(x,y)] = \Pr\left[\sum_{i=1}^{N}\sum_{j=1}^{N} m(x-x_i, y-y_j)\right] =$$

$$\prod_{i=1}^{N}{}^{\otimes} \Pr[m(x,y)] \equiv \Pr[m(x,y)] \otimes_x \otimes_y \Pr[m(x,y)] \otimes_x \otimes_y \ldots$$

$$\sim \text{Gaussian}(z), \quad N \to \infty$$

where Pr denotes the Probability Density Function. We can thus expect $\Pr[c(x,y)]$ to be normally distributed and for $m(x,y) \in [0,1] \forall x,y$ the mode of the distribution will be of the order of 0.5. This result provides a value for the threshold T in equation (2) which for $0 \leq c(x,y) \leq 1$ is 0.5 (theoretically). Note that if $n(x,y)$ is uniformly distributed and thereby represents δ-uncorrelated noise then both the complex spectrum N^* and power spectrum $\mid N \mid^2$ will also be δ-uncorrelated and since

$$m(x,y) = \mathcal{F}_2^{-1}\left[\frac{N^*(k_x,k_y)}{\mid N(k_x,k_y) \mid^2}\right]$$

$\Pr[m(x,y)]$ will be uniformly distributed. Also note that the application of a threshold which is given by the mode of the Gaussian distribution, guarantees that there is no statistical bias associated with any bit in the binary output, at least, on a theoretical basis. On a practical basis, the needs to be computed directly by calculating the mode from the histogram of the cipher and that bit equalization can not be guaranteed as it will depend on: (i) the size of the images used; (ii) the number of bins used to compute the histogram.

C. Principal Algorithms

The principal algorithms associated with the application of stochastic diffusion for watermarking with ciphers are as follows:

Algorithm I: Encryption and Watermarking Algorithm

Step 1: Read the binary plaintext image from a file and compute the size $I \times J$ of the image.

Step 2: Compute a cipher of size $I \times J$ using a private key and pre-condition the result.

Step 3: Convolve the binary plaintext image with the pre-conditioned cipher and normalise the output.

Step 4: Binarize the output obtained in Step 3 using a threshold based on computing the mode of the Gaussian distributed ciphertext.

Step 5: Insert the binary output obtained in Step 4 into the lowest 1-bit layer of the host image and write the result to a file.

The following points should be noted:

(i) The host image is taken to be an 8-bit or higher grey level image which must ideally be the same size as the plaintext image or else resized accordingly. However, in resembling the host image, its proportions should be the same so that the stegotext image does not appear to be a distorted version of the covertext image. For this purpose, a library of host images should be developed whose dimensions are set according to a predetermined application where the dimensions of the plaintext image are known.

(ii) Pre-conditioning the cipher and the convolution processes are undertaken using a Discrete Fourier Transform (DFT).

(iii) The output given in Step 3 will include negative floating point numbers upon taking the real component of a complex array. The array must be rectified by adding the largest negative value in the output array to the same array before normalisation.

(iv) For colour host images, the binary ciphertext can be inserted in to one or all of the RGB components. This provides the facility for watermarking the host image with three binary ciphertexts (obtained from three separate binary documents, for example) into a full colour image. In each case, a different key can be used.

(v) The binary plaintext image should have homogeneous margins in order to minimise the effects of ringing due to 'edge-effects' when processing the data in the spectral domain.

Algorithm II: Decryption Algorithm

Step 1: Read the watermarked image from a file and extract the lowest 1-bit layer from the image.

Step 2: Regenerate the (non-preconditioned) cipher using the same key used in Algorithm I.

Step 3: Correlate the cipher with the input obtained in Step 1 and normalise the result.

Step 4: Quantize and format the output from Step 3 and write to a file.

The following points should be noted:

(i) The correlation operation should be undertaken using a DFT.

(ii) For colour images, the data is decomposed into each RGB component and each 1-bit layer is extracted and correlated with the appropriate cipher, i.e. the same cipher or three ciphers relating to three private keys respectively.

(iii) The output obtained in Step 3 has a low dynamic range and therefore requires to be quantized into an 8-bit image based on floating point numbers within the range max(array)-min(array).

D. StegoText

StegoText is a prototype tool designed using MATLAB to examine the applications to which stochastic diffusion can be used. A demonstration version of the system is available at http://eleceng.dit.ie/arg/downloads/Stegocrypt which has been designed with a simple Graphical User Interface as shown in Figure 3 whose use is summarised in the following table:

Encryption Mode	**Decryption Mode**
Inputs: Plaintext image Covertext image Private Key (PIN)	*Inputs:* Stegotext image Private key (PIN)
Output: Watermarked image	*Output:* Decrypted watermark
Operation: Encrypt by clicking on buttom E (for Encrypt)	*Operation:* Decrypt by clicking on button D (for Dycrypt)

The PIN (Personal Identity Number) can be an numerical string with upto 16 elements. In principal, any existing encryption algorithm, application or system can be used to generate the cipher required by *StegoText* by encrypting an image composed of random noise. The output is then needs to be converted into a decimal integer array and the result normalised as required, i.e. depending on the format of the output that is produced by a given system. In this way, *StegoText* can be used in conjunction with any existing encryption standard.

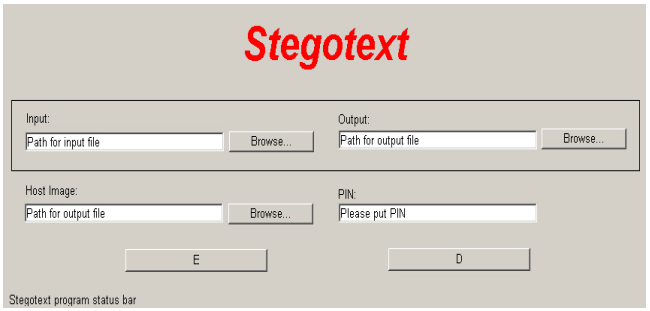

Fig. 3. Graphical User Interface for *Stegotext* software system.

The principal aim of *StegoText* is to encrypt an image and transform the ciphertext into a binary array which is then used to watermark a host image. This provides a general method for hiding encrypted information in 'image-space'.

E. e-Fraud Prevention of e-Certificates

Electronic or E-documents consisting of letters and certificates, for example, are routinely used in EDI. EDI refers to the structured transmission of data between organizations by electronic means. It is used to transfer electronic documents from one computer system to another; from one trading partner to another trading partner, for example [84]. The USA National Institute of Standards and Technology defines EDI as *the computer-to-computer interchange of strictly formatted messages that represent documents other than monetary instruments* [85]. EDI remains the data format used by the vast majority of electronic transactions in the world and EDI documents generally contain the same information that would normally be found in a paper document used for the same organizational function.

In terms of day-to-day applications, EDI relates to the use of transferring documents between two parties in terms of an attachment. For hardcopies, the attachment is typically the result of scanning the document and generating an image which is formatted as a JPEG or PDF (Print Device File) file, for example. This file is then sent as an attachment to an email which typically refers to the attachment, i.e. the email acts as a covering memorandum to the information contained in the attachment. However, a more common approach is to print a document directly to PDF file, for example. Thus, letters written in MicroSoft word, for example, can be routinely printed to a PDF file for which there are a variety of systems available, e.g. PDF suite http://pdf-format.com/suite/.

For letters and other documents that contain confidential information, encryption systems are often used to secure the document before it is attached to an email and sent. The method discussed in this paper provides a way of encrypting a document using stochastic diffusion and then hiding the output in an image, thus providing a covert method of transmitting encrypted information. However, the approach can also be used to authenticate a document by using the original document as a 'host image'. In terms of the *Stegotext* GUI shown in Figure 3, this involves using the same file for the *Input* and *Host Image*. An example of this is shown in Figure 4 where a hardcopy

issue of a certificate has been scanned into electronic form and the result printed to a PDF file. The properties of the image are as follows: File size=2.58Mb; Pixel Dimensions - Width=783 pixels, Height =1151 pixels; Document Size - Width=19.89 cm, Height=29.24cm; Resolution=100 pixels/inch. The result has been encrypted and binarised using stochastic diffusion and the output used to watermark the original document. The fidelity of the decrypt is perfectly adequate to authenticate aspects of the certificate such as the name and qualification of the holder, the date and signature, for example. Figure 5 shows the university stamp and signature associated with this decrypt which have been cut from the original decrypt given in Figure 4. These results illustrate that the decrypt is adequately resolved for the authentication of the document as a whole. It also illustrates the ability for the decrypt to retain the colour of the original plaintext image.

Fig. 4. Certificate with binary watermark (left) and decrypt (right).

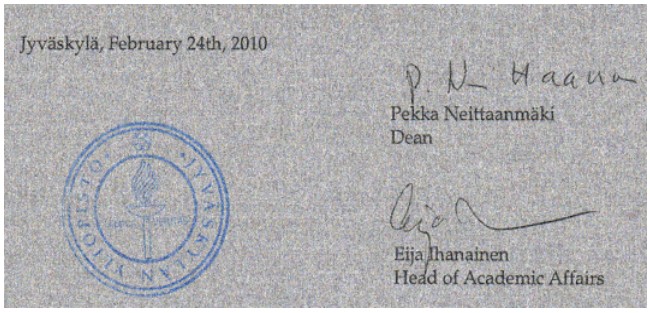

Fig. 5. 'Stamp' (left) and signatures (right) of decrypt given in Figure 4.

VI. Loss-less Encrypted Image Information Hiding

The method discussed in the previous section is suitable for document authentication, but the lossy nature of the reconstruction generated through binarisation of the cipher, illustrated in Figure 4, is not suitable for full colour images. In this section we introduce an algorithms for hiding grey scale image in a colour image and full colour images using three host colour images. Figure 6 shows a block diagram for hiding an encrypted 8-bit grey level image in a 24-bit colour image and Figure 7 shows the equivalent block diagram for hiding encrypted 24-bit colour image in three 24-bit colour host images. In the latter case, the same approach is used applied to each colour component of the colour image. Referring to Figure 6, stochastic diffusion is used to encrypt an 8-bit grey level image into a 24-bit colour host image with a near perfect decrypt. In this scheme, the cipher is not binarised but is converted into binary form. The first and second Least Significant Bits (LSBs) are ignored and the third and fourth bits are embedded into the two LSBs of the host image's red channel. Similarly, the 5^{th} and 6^{th} bits are embedded into the two LSBs of the host image's green channel, and finally the 7^{th} and 8^{th} bits are embedded into the two LSBs of the host image's blue channel. The inverse process is based on extracting the relevant bits from the associated channels with the first and second bits being set to zero. The extracted bits are then used to re-generate the original cipher and the reconstruction obtained by correlation with the original noise field.

Figure 8 shows an example of the method based on the block diagram given in Figure 7 using the MATLAB code given in Appendix C. The three 24-bit colour host images after application of the embedding process are given in Figure 9.

VII. Conclusions

In this paper, the concepts of cryptography and information hiding have been introduced and the use of these techniques for protecting important information is generally discussed. The use of encrypted information hiding and a range of methods have been studied and reported in an extensive survey. These methods are divided into two categories; the first category focuses on embedding encrypted data in the spatial domain of an image whereas the second category is based on the use of the transform domain to hide encrypted information.

The use of cryptographic algorithms together with steganography and watermarking methods make it almost impossible for attack to be launched in order to recover the encrypted hidden data because this requires the attacker to first detect the existence of the information before an attempt can be made to decrypt it. This provides an important extension to cryptography alone in that the incorporation of steganography and watermarking algorithms allows the existence of the encrypted data to be unknown. To recover the information the attacker needs to first find a way of extracted the hidden encrypted information from the covertext and then decrypting using the appropriate algorithm/key(s). The exposure of the encryption key(s), the encryption algorithm and the embedding technique (and the key(s) used for this process as required) to those other than the intended receiver is practically impossible.

A study of the methods reported in Section III and IV show that use of the transform domain, whether to encrypt the information and/or embed it is more secure and robust against attacks despite the complexity of such techniques. Encrypted information hiding methods that make use of the spatial domain tend to be less robust but easier to implement. In both cases, the techniques and algorithms developed have a wide range of applications with regard to keeping transmitted

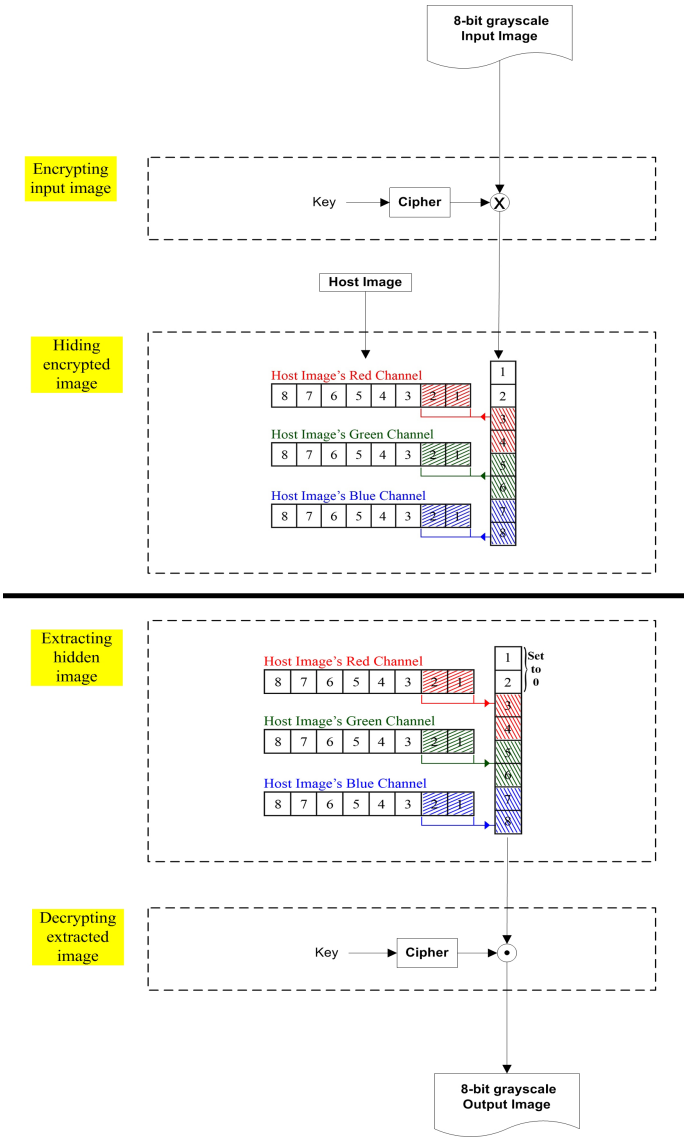

Fig. 6. Block Diagram for hiding an encrypted 8-bit grey level image in a 24-bit colour host image.

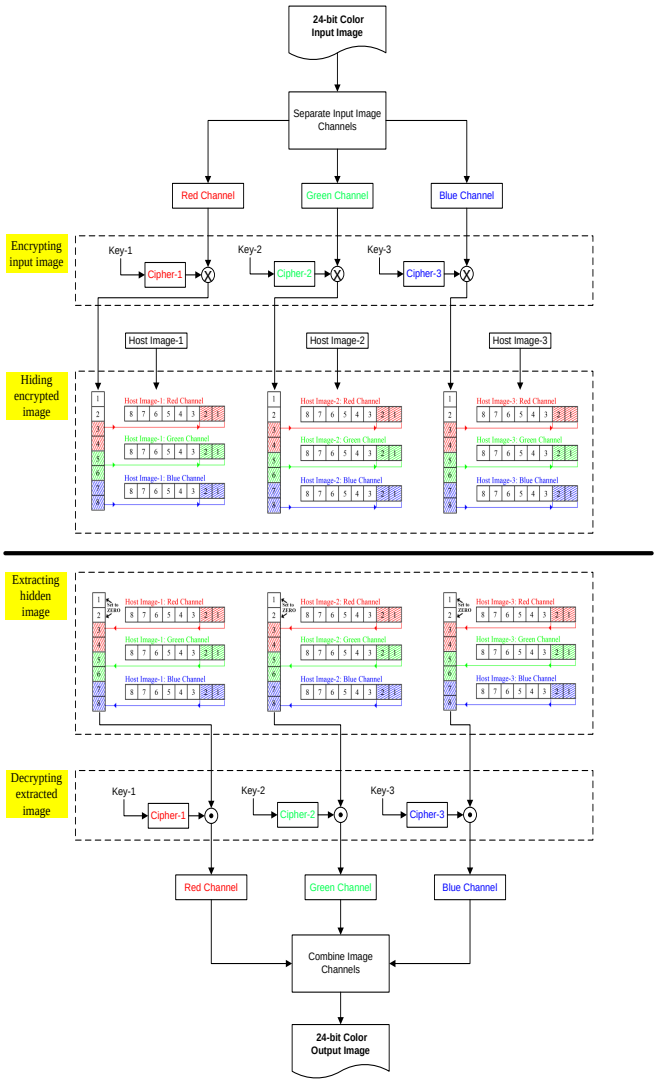

Fig. 7. Block Diagram for hiding an encrypted 24-bit colour image in three 24-bit colour host images.

data (principally image based data) both secure and covert. Authentication of digital data and documents, e-to-e data transmission, covert communication, copyright protection and tracking of the original source of information are further examples of areas where the concept of encrypted information hiding can be applied.

We have considered the application of stochastic diffusion for encrypting image information and hiding it in a host image. Embedding a binary watermark into a host image obtained by binarizing a floating point ciphertext, as discussed in Section V(A), provides a cryptographically secure solution. This is because binarization is an entirely one-way process. Thus, although the watermark may be removed from the covertext image, it can not be decrypted without the recipient having access to the correct cryptographically secure algorithm and key. This is the principal basis for the *StegoText* system reported in Section V(D) that is currently available and has a range of application for e-document authentication. For example, many institutes such as universities still issue 'paper certificates' to their graduates. These certificates are then scanned and sent as attachments along with a CV and covering letter when applying for a job. It is at this point that the certificate may be counterfeited and, for this reason, some establishments still demand originals to be submitted. *StegoText* provides the facility to issue electronic certificates (in addition or in substitution to a hardcopy) which can then be authenticated as discussed in Section V(E). By including a serial number on each certificate (a Certificate Identity Number) which represents a 'public key', the document can be submitted to the authority that issued the certificate for authentication, for which an online service can be established as required subject to any regulation of investigatory powers e.g. [86]. However, the principal focus of this paper has been to extend the application of stochastic diffusion to hide 24-bit

Fig. 8. Orginal Image (above) and reconstructed image after decryption (below).

colour images in a set of three 24-bit colour images to provide a high fidelity decrypt. This provides a near lossless method of encrypting and covertly communicating 24-bit colour images over the Internet as required as discussed in Section VI. The applications of this approach are numerous. Coupled with appropriate key-exchange protocols to initiate cryptographically strong ciphers, the approach provides a generic method of encrypting and hiding high fidelity digital image information.

APPENDIX A: MATLAB CODE FOR LOSS-LESS WATERMARKING METHOD

```
function []  = CIE( ImageName )
% This function - Covert Image Encryption
% (CIE) - inputs a 24-bit color image and
% encrypts it using Stochastic Diffusion.

% Read input image
```

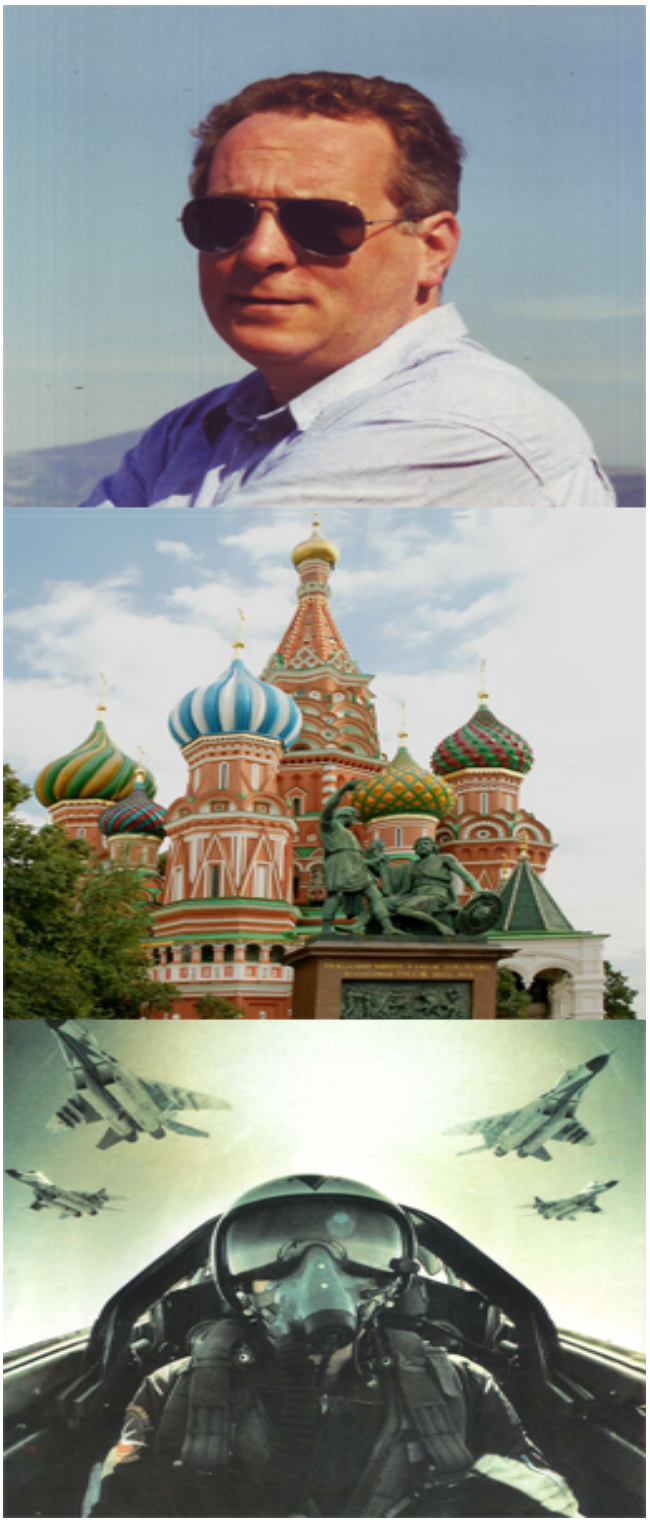

Fig. 9. Host images used to hide the image given in Figure 8 after embedding the ciphers.

```
InImage = imread(ImageName);
row = size(InImage,1);
col = size(InImage,2);
InImage = double(InImage);
%-------------------------

% Generate the noise field
% using Matlab's rand function
NoiseImageR = rand(row,col);
NoiseImageG = rand(row,col);
NoiseImageB = rand(row,col);

NR = NoiseImageR;
NG = NoiseImageG;
NB = NoiseImageB;
% -------------------------

% Convolve the input image with the
% noise image using a 2D FFT
% with pre-conditioning
mR = PreCondition(NoiseImageR);
mG = PreCondition(NoiseImageG);
mB = PreCondition(NoiseImageB);
% ----------------------------
% Encrypt the Red Channel
CR=ifft2(fft2(mR).*fft2(InImage(:,:,1)));
% Encrypt the Green Channel.
CG=ifft2(fft2(mG).*fft2(InImage(:,:,2)));
% Encrypt the Blue Channel
CB=ifft2(fft2(mB).*fft2(InImage(:,:,3)));

% Normalize Cipher Images to range 0:255.
CR = Normalize(CR) .* 255;
CG = Normalize(CG) .* 255;
CB = Normalize(CB) .* 255;
% ------------------------
CR = uint8(CR);
CG = uint8(CG);
CB = uint8(CB);
% ------------------------

% Embed cipher images into three
% named cover images:
%
% cover1.bmp
% cover2.bmp
% cover3.bmp

% Embed red channel cipher into
% cover image 1
CoverImage1 = imread('cover1.bmp');
CoverImage1 =
imresize(CoverImage1 , [row col]);
figure(1);
subplot(1,2,1), imshow(CoverImage1),
title('Cover Image1 before embedding');
for i = 1 : size(CoverImage1,1)
    for j = 1 : size(CoverImage1,2)

        CoverImage1(i,j,1) =
        bitand( CoverImage1(i,j,1) , 252 );

        CoverImage1(i,j,1) =
        bitor( CoverImage1(i,j,1),
        bitand(bitshift(CR(i,j),-2),3) );

        CoverImage1(i,j,2) =
        bitand( CoverImage1(i,j,2) , 252 );

        CoverImage1(i,j,2) =
        bitor( CoverImage1(i,j,2),
        bitand(bitshift(CR(i,j),-4),3) );

        CoverImage1(i,j,3) =
        bitand( CoverImage1(i,j,3) , 252 );
        CoverImage1(i,j,3) =
        bitor( CoverImage1(i,j,3),
        bitand(bitshift(CR(i,j),-6),3));

    end
end
subplot(1,2,2), imshow(CoverImage1),
title('Cover Image1 after Embedding');
% ----------------------------------
% Embed green channel cipher
% into Cover Image 2
CoverImage2 = imread('cover2.bmp');
CoverImage2 =
imresize(CoverImage2 , [row col]);
figure(2);
subplot(1,2,1), imshow(CoverImage2),
title('Cover Image2 before Embedding');

for i = 1 : size(CoverImage2,1)
    for j = 1 : size(CoverImage2,2)

        CoverImage2(i,j,1) =
        bitand( CoverImage2(i,j,1) , 252 );
        CoverImage2(i,j,1) =
        bitor( CoverImage2(i,j,1),
        bitand(bitshift(CG(i,j),-2),3) );

        CoverImage2(i,j,2) =
        bitand( CoverImage2(i,j,2) , 252 );
        CoverImage2(i,j,2) =
        bitor( CoverImage2(i,j,2),
        bitand(bitshift(CG(i,j),-4),3) );

        CoverImage2(i,j,3) =
        bitand( CoverImage2(i,j,3) , 252 );
        CoverImage2(i,j,3) =
        bitor( CoverImage2(i,j,3),
        bitand(bitshift(CG(i,j),-6),3));

    end
end
```

```
subplot(1,2,2), imshow(CoverImage2),
title('Cover Image2 after Embedding');
% ------------------------------------
% Embed blue channel cipher into
% Cover Image 3
CoverImage3 = imread('cover3.bmp');
CoverImage3 =
imresize(CoverImage3 , [row col]);

figure(3);
subplot(1,2,1), imshow(CoverImage3),
title('Cover Image3 before Embedding');

for i = 1 : size(CoverImage3,1)
    for j = 1 : size(CoverImage3,2)

    CoverImage3(i,j,1) =
    bitand( CoverImage3(i,j,1) , 252 );

    CoverImage3(i,j,1) =
    bitor( CoverImage3(i,j,1),
    bitand(bitshift(CB(i,j),-2),3) );

    CoverImage3(i,j,2) =
    bitand( CoverImage3(i,j,2) , 252 );
    CoverImage3(i,j,2) =
    bitor( CoverImage3(i,j,2),
    bitand(bitshift(CB(i,j),-4),3) );

    CoverImage3(i,j,3) =
    bitand( CoverImage3(i,j,3) , 252 );
    CoverImage3(i,j,3) =
    bitor( CoverImage3(i,j,3),
    bitand(bitshift(CB(i,j),-6),3));

    end
end
subplot(1,2,2), imshow(CoverImage3),
title('Cover Image3 after Embedding');
% ------------------------------------

% Extract the hidden ciphers from
% cover images
%
% Extract red channel cipher from
% cover image 1
for i = 1 : size(CoverImage1,1)
    for j = 1 : size(CoverImage1,2)

    R = bitand( CoverImage1(i,j,1), 3);
    G = bitand( CoverImage1(i,j,2), 3);
    B = bitand( CoverImage1(i,j,3), 3);
    ExImageR(i,j) =
    bitor( bitor(bitshift(R,2),

    bitshift(G,4)),
    bitshift(B,6) );

    end
end
ExImageR = uint8(ExImageR);
% ------------------------
%
% Extract green channel cipher
% from cover image 2
for i = 1 : size(CoverImage2,1)
    for j = 1 : size(CoverImage2,2)

    R = bitand( CoverImage2(i,j,1), 3);
    G = bitand( CoverImage2(i,j,2), 3);
    B = bitand( CoverImage2(i,j,3), 3);
    ExImageG(i,j) =
    bitor( bitor(bitshift(R,2),

    bitshift(G,4)),
    bitshift(B,6) );

    end
end
ExImageG = uint8(ExImageG);
% ------------------------
%
% Extract blue channel cipher
% from cover image 3
for i = 1 : size(CoverImage3,1)
    for j = 1 : size(CoverImage3,2)

    R = bitand( CoverImage3(i,j,1), 3);
    G = bitand( CoverImage3(i,j,2), 3);
    B = bitand( CoverImage3(i,j,3), 3);
    ExImageB(i,j) =
    bitor( bitor(bitshift(R,2),

    bitshift(G,4)),
    bitshift(B,6) );

    end
end
ExImageB = uint8(ExImageB);
% ------------------------

% Correlate the extracted ciphers
% with the noise field using a 2D FFT
ExImageR = double(ExImageR);
ExImageG = double(ExImageG);
ExImageB = double(ExImageB);

PlainImR =
ifft2(conj(fft2(NR)) .* fft2(ExImageR));

PlainImG =
ifft2(conj(fft2(NG)) .* fft2(ExImageG));

PlainImB =
ifft2(conj(fft2(NB)) .* fft2(ExImageB));
```

```
% Normalize images to raneg 0:255
PlainImR = Normalize(PlainImR) .* 255;
PlainImG = Normalize(PlainImG) .* 255;
PlainImB = Normalize(PlainImB) .* 255;
%-----------------------------------

Result(:,:,1) = PlainImR;
Result(:,:,2) = PlainImG;
Result(:,:,3) = PlainImB;
Result = uint8(Result);
imwrite(Result,'Output_Color.bmp');

figure(4);
subplot(1,2,1), imshow(uint8(InImage)),
title('Input Image before Encryption');

subplot(1,2,2), imshow(Result),
title('Output Image after Decryption');

end

%-----------------------------------

function [ x ] = Normalize( mat )
% Function to normalise images

MAX = max(mat(:)); MIN = min(mat(:));

for i = 1:size(mat,1)
    for j = 1:size(mat,2)
x(i,j) = ((mat(i,j) - MIN )/(MAX - MIN));
    end
end

return;
end

%---------------------------------

function [ m ] = PreCondition( arr )
% Pre-conditioning function

arrF = fft2(arr);
for i = 1:size(arrF,1)
    for j = 1:size(arrF,2)
        if abs(arrF(i,j)) == 0
            M(i,j) = arrF(i,j);
        else

    M(i,j)=arrF(i,j)/(abs(arrF(i,j))
           *abs(arrF(i,j)));

        end
    end
end
m = ifft2(M);

return;
end
```


ACKNOWLEDGMENT

The work reported in this paper is funded by the Science Foundation Ireland. The authors are grateful for the support of Dublin Institute of Technology, School of Electrical Engineering Systems.

REFERENCES

[1] W. Stallings, *Cryptography and Network Security: Principles and Practice*, New Jersey: Prentice-Hall, 1999.

[2] B. Schneier, *Applied Cryptography (Second Edition)*, Wiley, NewYork, 1996.

[3] National Bureau of Standards, Data Encryption Standard, FIPS Publication 46, US Department of Commerce, 1977.

[4] R. L. Rivest, A. Shamir and L. Adleman, *A Method for Obtaining Digital Signatures and Public Key Cryptosystems*, Magazine Communication of the ACM, 120-126, 1978.

[5] Y. H. Chu and S. Chang, *Dynamical Cryptography based on Synchronized Chaotic Systems*, Electronics Letters, 974-975, 1999.

[6] H. J. Highland, *Data Encryption: A Non-mathematical Approach*, Computer Security, 369-386, 1997.

[7] P. Refregier, B. Javidi, *Optical Image Encryption using Input and Fourier Plane Random Phase Encoding*, Journal of Optics Letters, vol. 20, 767-769, 1995.

[8] G. Unnikrishnan, J. Joseph and K. Singh, *Optical Encryption by Double-Random Phase Encoding in the Fractional Fourier Domain*, Optics Letters, Vol. 25, Issue 12, 887-889, 2000.

[9] B. H. Zhu, S. T. Liu and Q. W. Ran, *Optical Image Encryption based on Multi-fractional Fourier Transforms*, Optics Letters, 1159-1161, 2000.

[10] R. Tao, Y. Xin and Y. Wang, *Double Image Encryption based on Random Phase Encoding in the Fractional Fourier Domain*, Optics Letters, 16067-79, 2000.

[11] G. Situ and J. Zhang, *Double Random-Phase Encoding in the Fresnel Domain*, Optics Letters, Vol. 29, Issue 14, 1584-1586, 2004

[12] L. Yu, X. Peng and L.Cai, *Parameterized Multi-dimensional Data Encryption by Digital Optics*, Optics Communications, 67-77, 2002.

[13] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C (Second Edition)*, Wiley, 1996.

[14] C. P. Pfleeger and S. L. Pfleeger, *Security in Computing (Fourth Edition)*, Prentice Hall, 2006.

[15] I. Yin-Nii and K. Bao-Sheng, *Digital Image Scrambling based on Improved Arnold Transformation*, Computer Technology and Development, Vol. l8, No.2, 74-76, 2008.

[16] G. Xinke and M. A. Xianqing, *Image Digital Watermarking Technology based on Image-scrambling and the DWT*, Science Technology and Engineering, Vol. 8, No. 19, 5412-5416, 2008.

[17] C. Ping and Z. Fei, *Image Encryption Algorithm based on Arnold Image-Scrambling and the Wavelet Transform*, Microelectronics and Computers, No.lO, 197-199, 2009.

[18] L. I. Hao, L. V. Jianping and Y. Fangfang, *Wavelet Domain Digital Image Watermarking Algorithm based on Scrambling Encryption*, Journal of Telecommunications, Vol. l4, No. 5, 107-11O, 2009.

[19] S. Qiu-dong, M. A. Wen-xin, Y. A. Wen-ying and D. Hong, *A Dual Random Scrambling Method for Digital Image Encryption Based on the Wavelet Transform*, Journal of Shanghai Second Polytechnic University, No.4, 1-5, 2008.

[20] J. M. Blackledge, *Multi-algorithmic Cryptography using Deterministic Chaos with Application to Mobile Communications*, ISAST Transactions on Electronics and Signal Processing, Vol. 1, No. 2, 23-64, 2008.

[21] J. W. Yoon and H. Kim, *An Image Encryption Scheme with a Pseudorandom Permutation based on Chaotic Maps*, Communications in Nonlinear Science and Numerical Simulation, Vol. 15, Issue 12, 3998-4006, 2010.

[22] S. Yang and S. Sun, *A Video Encryption Method based on Chaotic Maps in the DCT Domain*, Progress in Natural Science, Vol. 18, Issue 10, 1299-1304, 2008.

[23] M. Long and L. Tan, *A Chaos-Based Data Encryption Algorithm for Image/Video*, IEEE Conferences, 172-175, 2010.

[24] Y. Chen, L. Zhang and Y. Weng, *A Data Encryption Algorithm based on Dual Chaotic Systems*, IEEE Conferences, V4-431 - V4-435, 2010.

[25] H. Liu and X. Wang, *Color Image Encryption based on One-time Keys and Robust Chaotic Maps*, Computers and Mathematics with Applications, Vol. 59, Issue 10, 3320-3327, 2010.

[26] D. Bucerzan, *A Cryptographic Algorithm Based on a Pseudorandom Number Generator*, IEEE Conferences, V453-456, 2008.

[27] J. W. Yoona and H. Kim, *An Image Encryption Scheme with a Pseudorandom Permutation based on Chaotic Maps*, Communications in Nonlinear Science and Numerical Simulation, Vol. 15, Issue 12, 3998-4006, 2010.

[28] I. Cox, M. Miller, J. Bloom, J. Fridrich and T. Kalker, *Digital Watermarking and Steganography (Second Edition)*, Morgan- Kaufmann, 2007.

[29] F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn, *Information Hiding: A Survey*, Proceedings of the IEEE, 1062-1078, 1999.

[30] http://en.wikipedia.org/wiki/Steganography.

[31] G. Abboud, J. Marean and R. V. Yampolskiy, *Steganography and Visual Cryptography in Computer Forensics*, IEEE Conferences, 25-32, 2010.

[32] S. Katzenbeisser, F. A. Petitcolas, *Information Hiding Techniques for Steganography and Digital Watermarking* , Artech House Inc, London, 2000.

[33] K. W. Mahmoud, *Novel Methods for Print Security and Anti-counterfeiting Technology*, PhD Thesis, Loughborough University, 2006.

[34] J. M. Blackledge and D. Dubovitski, *Covert Encryption Method for Applications in Electronic Data Interchange*, ISAST Trans. on Electronics and Signal Processing, Vol. 4, Issue 1, 107-128, 2009.

[35] J. M. Blackledge and E. Coyle, *e-Fraud Prevention based on the Self-Authentication of e-Documents*, IEEE Computer Society, The Fourth International Conference on Digital Society, St. Maarten, Netherlands Antilles, Vol. 978-0-7695-3953-9, 329 - 338, 2010.

[36] X. Li, Z. Qi, Z. Yang and J. Kong, *A Novel Hidden Transmission of Biometric Images based on Choas and Image Content*, First International Workshop on Education Technology and Computer Science, 21-25, 2009.

[37] J. Kong, H. Jia, X. Li and Z. Qi , *A Novel Content-based Information Hiding Scheme*, International Conference on Computer Engineering and Technology, 436-440, 2009.

[38] C. W. Lee and W. H. Tsai, *A New Steganographic Method Based on Information Sharing via PNG Images*, 2nd International Conference on Computer and Automation Engineering (ICCAE), 807-811, 2010.

[39] C. Uuefen, L. Junhuan, Z. Shiqing and C. Caiming, *Double Random Scrambling Algorithm based on Subblocks for Image Hiding*, International Conference on Computer and Communication Technologies in Agriculture Engineering, 255-257, 2010.

[40] M. K. Kundu and S. Das, *Lossless ROI Medical Image Watermarking Technique with Enhanced Security and High Payload Embedding*, International Conference on Pattern Recognition, 1457-1460, 2010.

[41] Y. Lu, X. Li, M. Qi, J. Li, Y. Fu and J. Kong, *Lossless and Content-based Hidden Transmission for Biometric Verificaion*, Second International Symposium on Intelligent Information Technology Application, 462-466, 2008.

[42] M.Sabery and Y. Yaghoobi, *A Simple and Robust Approach for Image Hiding using Chaotic Logistic Map*, International Conference on Advanced Computer Theory and Engineering, 623-627, 2008.

[43] J. M. Blackledge, *Information Hiding using Stochastic Diffusion for the Covert Transmission of Encrypted Images*, Irish Signals and Systems Conference, 12-17, 2010.

[44] D. C. Lou and J. L. Liu, *Steganographic Method for Secure Communications*, Computers and Security, Vol. 21, Issue 5, 449-460, 2002.

[45] Z. Liu, M. A. Ahmad and S. Liu, *Image Sharing Scheme based on Combination Theory*, Optics Communications, Vol. 281, Issue 21, 5322-5325, 2008.

[46] S. C. Shie, S. D. Lin and J. H. Jiang, *Visually Imperceptible Image Hiding Scheme based on Vector Quantization*, Information Processing and Management, Vol. 46, Issue 5, 495-501, 2010.

[47] Y. Linde, A. Buzo and R. M. Gray, *An Algorithm for Vector Quantizer Design*, IEEE Transactions on Communications, Vol. 28, Issue 1, 84-95, 1980.

[48] Y. H. Yu, C. C. Chang and I. C. Lin, *A New Steganographic Method for Color and Grayscale Image Hiding*, Computer Vision and Image Understanding, Vol. 107, Issue 3, 183-194, 2007.

[49] Y. H. Yu, C. C. Chang and Y. C. Hu, *Hiding Secret Data in Images via Predictive Coding*, Pattern Recognition, Vol. 38, Issue 5, 691-705, 2005.

[50] J. Jiang, B. Guo and S. Y. Yang, *Revisiting the JPEG-LS Prediction Scheme*, IEEE Proceedings-Vision, Image and Signal Processing, Vol. 147, Issue 6 575-580, 2000.

[51] C. C. Chang, C. S. Chan and Y. H. Fan, *Image Hiding Scheme with Modulus Function and Dynamic Programming Strategy on Partitioned Pixels*, Pattern Recognition, Vol. 39, Issue 6, 1155-1167, 2006.

[52] S. J. Wang, *Steganography of Capacity Required using Modulo Operator for Embedding Secret Images*, Applied Mathematics and Computation, Vol. 164, Issue 1, 99-116, 2005.

[53] Y. S. Wu, C. C. Thien and J. C. Lin, *Sharing and Hiding Secret Images with Size Constraint*, Pattern Recognition, Vol. 37, Issue 7, 1377-1385, 2004.

[54] C. C. Thien and J. C. Lin, *Secret Image Sharing*, Computers and Graphics, Vol. 26, Issue 5, 765-770, 2002.

[55] L. Chen, D. Zhao and F. Ge, *Gray Images Embedded in a Color Image and Encrypted with FRFT and Region Shift Encoding Methods*, Optics Communications, Vol. 283, Issue 10, 2043-2049, 2010.

[56] X. F. Meng, L. Z. Cai, X. L. Yang, X. F. Xu, G. Y. Dong, X. X. Shen, H. Zhang and Y. R. Wang, *Digital Color Image Watermarking based on Phase-shifting Interferometry and Neighboring Pixel Value Subtraction Algorithm in the DCT Domain*, Applied Optics, Vol. 46, Issue 21, 4694-4701, 2007.

[57] X. Zhou and J. G. Chen, *Information Hiding based on Double-random Phase Encoding Technology*, J. Mod. Opt., Vol. 53, Issue 12, 1777-1783, 2006.

[58] X. Hou, D Lai, S. Yuan, D. H. Li and J. P. Hu, *A Method for Hiding Information utilizing Double-random Phase-encoding Techniques*, Optics Laser Technology, Vol. 39, Issue 7, 1360-1363, 2007.

[59] K. T. Lin, *Digital Information Encrypted in an image using Binary Encoding*, Optics Communications, Vol. 281, Issue 13, 3447-3453, 2008.

[60] J. M. Blackledge and K. Mahmoud, *Printed Document Authentication using Texture Coding*, ISAST Transaction on Electronics and Signal Processing, Vol. 4, Issue 1, 81-98, 2009.

[61] J. M. Blackledge, *Authentication of Biometric Features using Texture Coding for ID Cards*, IEEE Computer Society, The Fifth International Conference on Internet Monitoring and Protection , Barcelona, Spain, Vol. 978-0-7695-4023-8, 74 - 83, 2010.

[62] D. W. Kim, H. J. Choi, Y. G. Choi, J. S. Yoo and Y. H. Seo, *Information Hiding for Digital Holograms by Electronic Partial Encryption Methods*, Optics Communications, Vol. 277, Issue 2, 277-287, 2007.

[63] W. Na, Z. Chiya, L. Xia and W. Yunjin, *Enhancing Iris-Feature Security with Steganography*, The fifth IEEE Conference onIndustrial Electronics and Applications (ICIEA), 2233-2237, 2010.

[64] C. C. Chang, T. S. Chen and L. Z. Chung, *A Steganographic Method based upon JPEG and Quantization Table Modification*, Information Sciences, Vol. 141, Issues 1-2, 123-138, 2002.

[65] C. T. Hsu and J. L. Wu, *Hidden Digital Watermarks in Images*, IEEE Transactions on Image Processing, Vol. 8, Issues 1, 58-68, 1999.

[66] J. Panada, J. Bisht, R. Kapoor and A. Bhattacharyya, *Digital Image Watermarking in Integer Wavelet Domain using Hybrid Techniques*, International Conference on Advances in Computer Engineering (ACE), 163-167, 2010.

[67] Z. Fanm, S. Dongfang and W. Yujing, *Adapting Digital Watermark Algorithm based on Choas and Image Fusion*, Global Congress on Intelligent Systems, Vol. 8, 126-130, 2009.

[68] Z. Wei, C. Zhi-gang and C. Yue-li, *Image Data Encryption and Hiding based on Wavelet Packet Transform and Bit Planes Decomposition*, Fourth International Conference on Wireless Communications, Networking and Mobile Computing, 1-4, 2008.

[69] J. J. Wang, B. L. Yang and Z. Yang, *An Algorithm of Information Hiding Based on Compound Encryption in Non-uniform B-spline Wavelet Domain*, Pacific-Asia Workshop on Computational Intelligence and Industrial Application, Vol. 2, 190-194, 2008.

[70] Q. D. Sun and J. C. Zuo, *A New Watermarking and Information Hiding Technique Based on Descrite Wavelet Transform*, First International Conference on Information Science and Engineering (ICISE), 1067-1070, 2009.

[71] M. K. Khan, J. Zhang and L. Tian, *Chaotic Secure Content-Based Hidden Transmission of Biometric Templates*, Chaos, Solitons and Fractals, Vol. 32, Issue 5, 1749-1759, 2007.

[72] K. J. Anil, S. Pankanti and R. Bolle, *Biometrics: Personal Identification in Networked Society*, Kluwer, 1999.

[73] J. Zhang, L. Tian and H. M. Tai, *A New Watermarking Method based on Chaotic Maps*, IEEE International Conference on Multimedia and Expo (ICME), Vol. 2, 939-942, 2004.

[74] D. Hitzl and F. Zele, *An Exploration of the Henon Quadratic Map*, Physica D: Nonlinear Phenomena, Vol. 14, Issue 3, 305-326, 1985.

[75] F. Ge, L. Chen and D. Zhao, *A Half-blind Color Image Hiding and Encryption Method*, Optics Communications, Vol. 281, Issue 17, 4254-4260, 2008.

[76] M. K. Khan, L. Xie and J. Zhang, *Chaos and NDFT-Based Spread Spectrum Concealing of Fingerprint-Biometric Data into Audio Signals*, Digital Signal Processing, Vol. 20, Issue 1, 179-190, 2010.
[77] K.J. Anil, S. Prabhakar, L. Hong and S. Pankanti, *Filter Bank-Based Fingerprint Matching*, IEEE Transactions on Image Processing, Volume 9, Issue 5, 846-859, 2000.
[78] K.J. Anil, S. Prabhakar and L. Hong, *A Multichannel Approach to Fingerprint Classification*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 21, Issue 4, 348-359, 1999.
[79] A. Tefas, N. Nikolaidis, V. Solachidis, S. Tsekeridou and I. Pitas, *Markov Chaotic Sequences for Correlation Based Watermarking Schemes*, Chaos, Solitons and Fractals, Vol. 17, Issues 2-3, 567-573, 2003.
[80] T. Anastasios, N. Athanasios, N. Nikolaidis, V. Solachidis, S. Tsekeridou and I. Pitas, *Performance Analysis of Correlation-Based Watermarking Schemes Employing Markov Chaotic Sequences*, IEEE Transactions on Signal Processing, Vol. 51, Issue 7, 1979-1994, 2003.
[81] B. Sklar, *Digital Communications: Fundamentals and Applications (Second Edition)*, Prentice Hall, 2001.
[82] D. He, H. Chen, L. G. Jiang, H. W. Zhu and G. R. Hu, *Chaotic Characteristics of One-Dimensional Iterative Maps with Finite Collapses*, IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, Vol. 48, Issue 7, 900-906, 2001.
[83] S. Chen and H. Leung, *Ergodic Chaotic Parameter MOdulation with Application to Digital Image Watermarking*, IEEE Transactions on Image Processing, Vol. 14, Issue 10, 1590-1602, 2005.
[84] http://en.wikipedia.org/wiki/Electronic_Data_Interchange
[85] M. Kantor and J. H. Burrows, *Electronic Data Interchange*, National Institute of Standards and Technology, 1996 http://www.itl.nist.gov/fipspubs/fip161-2.htm.
[86] http://www.opsi.gov.uk/acts/acts2000/ukpga_20000023_en_1

Jonathan Blackledge graduated in physics from Imperial College in 1980. He gained a PhD in theoretical physics from London University in 1984 and was then appointed a Research Fellow of Physics at Kings College, London, from 1984 to 1988, specializing in inverse problems in electromagnetism and acoustics. During this period, he worked on a number of industrial research contracts undertaking theoretical and computational research into the applications of inverse scattering theory for the analysis of signals and images. In 1988, he joined the Applied Mathematics and Computing Group at Cranfield University as Lecturer and later, as Senior Lecturer and Head of Group where he promoted postgraduate teaching and research in applied and engineering mathematics in areas which included computer aided engineering, digital signal processing and computer graphics. In 1994, Jonathan Blackledge was appointed Professor of Applied Mathematics and Head of the Department of Mathematical Sciences at De Montfort University where he expanded the post-graduate and research portfolio of the Department and established the Institute of Simulation Sciences. From 2002-2008 he was appointed Visiting Professor of Information and Communications Technology in the Advanced Signal Processing Research Group, Department of Electronics and Electrical Engineering at Loughborough University, England (a group which he co-founded in 2003 as part of his appointment). In 2004 he was appointed Professor Extraordinaire of Computer Science in the Department of Computer Science at the University of the Western Cape, South Africa. He currently holds the prestigious Stokes Professorship in ICT under the Science Foundation Ireland Programme based in the School of Electrical Engineering Systems, Dublin Institute of Technology and is Distinguished Professor at the Centre for Advanced Studies, Warsaw University of Technology, Poland.

AbdulRahman Isam Al-Rawi received his BSc in Computer Science from Zarka Private University in 2003 and an MSc in Computer Science from the University of Baghdad in 2006 specialising in image processing. After comletng his studies he took up academic appointments at the University of Baghdad and then at the University of Bahrain where he is currently a Lecturer in the College of Applied Studies. He is now undertaking a PhD at Dublin Institute of Technology, Ireland, under the supervision of Professor J M Blackledge with research interests that include information security, image processing, texture synthesis and editing, information hiding, digital watermarking and Steganography.